Micro-computer implementation of the Recursive Least Squares Parameter Estimation Algorithm with an example solution of the Deterministic Schaefer Equation

by R.V. Palermo

Department of Fisheries and Oceans
Field Services Branch
1090 West Pender Street,
Vancouver, B.C. V6E 2P1

February, 1984

Canadian Technical Report of
Fisheries and Aquatic Sciences
No. 1251

Canadian Technical Report of Fisheries and Aquatic Sciences

Technical reports contain scientific and technical information that contributes to existing knowledge but which is not normally appropriate for primary literature. Technical reports are directed primarily toward a worldwide audience and have an international distribution. No restriction is placed on subject matter and the series reflects the broad interests and policies of the Department of Fisheries and Oceans, namely, fisheries and aquatic sciences.

Technical reports may be cited as full publications. The correct citation appears above the abstract of each report. Each report is abstracted in *Aquatic Sciences and Fisheries Abstracts* and indexed in the Department's annual index to scientific and technical publications.

Numbers 1–456 in this series were issued as Technical Reports of the Fisheries Research Board of Canada. Numbers 457–714 were issued as Department of the Environment, Fisheries and Marine Service, Research and Development Directorate Technical Reports. Numbers 715–924 were issued as Department of Fisheries and the Environment, Fisheries and Marine Service Technical Reports. The current series name was changed with report number 925.

Technical reports are produced regionally but are numbered nationally. Requests for individual reports will be filled by the issuing establishment listed on the front cover and title page. Out-of-stock reports will be supplied for a fee by commercial agents.

Rapport technique canadien des sciences halieutiques et aquatiques

Les rapports techniques contiennent des renseignements scientifiques et techniques qui constituent une contribution aux connaissances actuelles, mais qui ne sont pas normalement appropriés pour la publication dans un journal scientifique. Les rapports techniques sont destinés essentiellement à un public international et ils sont distribués à cet échelon. Il n'y a aucune restriction quant au sujet; de fait, la série reflète la vaste gamme des intérêts et des politiques du ministère des Pêches et des Océans, c'est-à-dire les sciences halieutiques et aquatiques.

Les rapports techniques peuvent être cités comme des publications complètes. Le titre exact paraît au-dessus du résumé de chaque rapport. Les rapports techniques sont résumés dans la revue *Résumés des sciences aquatiques et halieutiques*, et ils sont classés dans l'index annual des publications scientifiques et techniques du Ministère.

Les numéros 1 à 456 de cette série ont été publiés à titre de rapports techniques de l'Office des recherches sur les pêcheries du Canada. Les numéros 457 à 714 sont parus à titre de rapports techniques de la Direction générale de la recherche et du développement, Service des pêches et de la mer, ministère de l'Environnement. Les numéros 715 à 924 ont été publiés à titre de rapports techniques du Service des pêches et de la mer, ministère des Pêches et de l'Environnement. Le nom actuel de la série a été établi lors de la parution du numéro 925.

Les rapports techniques sont produits à l'échelon régional, mais numérotés à l'échelon national. Les demandes de rapports seront satisfaites par l'établissement auteur dont le nom figure sur la couverture et la page du titre. Les rapports épuisés seront fournis contre rétribution par des agents commerciaux.

Canadian Technical Report of
Fisheries and Aquatic Sciences No.1251

February,1984

Micro-comouter implementation of the Recursive Least Squares Parameter ΞstimationAlgorithm with an example solution of the Deterministic Schaefer Equation

by

R.V. Palermo

Department of Fisheries and Oceans
Field Services Branch
1090 W. Pender Street
Vancouver, B.C.
V6E 2P1

(c) Minister of Supply and Services Canada 1984
Cat. No. FG 97-6/1251 ISSN 0706-6457

CONTENTS

List of Tables.....iv

Abstract/Resume'..... v

Introduction..... 1

Least Squares Analysis..... 1

The Recursive Least Squares Algorithm..... 4

The Program..... 6

Two Examples..... 7

Program Improvements..... 9

Conclusion..... 9

References.....9

Acknowledgements.....10

Listing A. Listing of Program for RLS Algorithm.....11

Listing B. Listing of RLS Output for Problem 1.....17

Listing C. Listing of RLS Output for Problem 2.....19

LIST OF TABLES

1. Sample data for the moving body problem.....16

2. RLS results for the moving body problem.....16

3. Effort data for the Hecate Strait Rock Sole.....18

ABSTRACT

Palermo, R.V.,1984. Micro-computer implementation of the Recursive Least Squares parameter estimation algorithm with an example solution of the deterministic Schaefer Equation, Can. Tech.Rep. Fish. Aquat. Sci 1251-vi-19p.

The method of Least Squares Analysis is a powerful analytical technique used in the analysis of observations. In this report, a Recursive Least Squares Algorithm is examined and its implementation on a micro-computer (Apple II+ or Apple IIe) is presented. The advantage of the recursive algorithm over classical Least Squares Analysis of multiple parameter problems lies in the fact that no matrix inversion is required. Therefore, the Recursive Algorithm is memory efficient, fast, and can be used sequentially and on-line to solve multiple parameter estimation problems. These feature allows its viable use on micro-computers. A further advantage of the recursive algorithm is that the parameter estimates are calculated at each sampling step, and a measure of their accuaracy is displayed in the variance-covariance matrix. The solution of a 3 parameter deterministic Schaefer model applied to data from the Hecate Strait Rock Sole is presented as an example of the Recursive Least Squares Algorithm application to a fisheries problem.

Key Words: Recursive Least Squares Algorithm, variance-covariance matrix, parameter estimation, micro-computer implementation, Schaefer model

Resume'

Title

Palermo,R.V., 1984. Micro-computer implementation of the Recursive least Squares parameter estimation algorithm with an example solution of the deterministic Schaefer Equation, Can. Tech.Rep. Fish. Aquat. Sci- 1251-vi-19p.

L'analyse par méthode des moindres carrés est une technique très efficace pour l'analyse des observations. On traite dans la présente communication d'un algorithme récurrent de moindres carrés et de son utilisation sur micro-ordinateurs (Apple II - et Apple IIe). La supériorité de l'algorithme récurrent par rapport à l'analyse classique, par moindres carrés, de problèmes à paramètres multiples réside dans l'élimination de la nécessité d'effectuer une inversion de matrice. L'algorithme récurrent réduit donc l'encombrement de la mémoire, est rapide et peut être utilisé séquentiellement ou en direct pour résoudre des problèmes d'estimation à paramètres multiples. Ces caractéristiques en permettent l'utilisation efficace sur micro-ordinateurs. Un autre avantage de cet algorithme réside dans le fait que les estimations des paramètres sont calculées à chaque étape d'échantillonnage et qu'une mesure de leur précision est affichée dans la matrice de variance-covariance. La solution d'un modèle déterministe de Schaefer à trois paramètres appliqué à des données sur la sole du Pacifique dans le détroit d'Hecate est présentée comme exemple de l'application de cet algorithme à un problème halieutique.

Mots clés: Algorithme récurrent de moindres carrés, matrice de variance-covariance, estimation de paramètres, utilisation sur micro-ordinateur, modèle de Scheafer.

Introduction

The method of Least Squares Analysis is a powerful analytical technique used in the analysis of observations. It's relative simplicity and ruggedness, coupled with wide applicability to various field of endeavours, has made it one of the most important tools for data analysis.

The intent of this article is to present a short introduction to Least Squares Analysis and a Recursive Least Square Algorithm. A program written in Applesoft Basic implementing the Recursive Least Squares Algorithm (RLS) is also presented.

Least Squares Analysis

Much scientific thought is concerned with the relationship between variables hpothesized to be in a cause-and-effect relationship, or in statistical jargon, a functional relationship. A function is a mathematical model enabling the prediction of some variable Y to one or more variables X. Such a relationship is written $Y = f(X)$. Y is called the dependent variable, and X (or X's) the independent variable(s). The magnitude of Y depends on the magnitude of X and can thus be predicted from the independent variable(s) which are free to vary.

Consider a linear regression problem in which a variable Y is known to be related to n other linearly independent variables X_j, $j = 1,2,3,\ldots n$, by a linear relationship of the form.

$$Y = a_1 x_1 + a_2 x_2 + \ldots + a_n x_n \qquad (1)$$

Where

a_j, $j = 1,2,\ldots n$ are n unknown but constant parameters which characterize the relationship and are to be estimated. The independent variables x_j are known, but Y is observed with error.

. . . /2

Then

$$Y = Y + \varepsilon Y$$

or

$$Y = a_1, x_1 + a_2 x_2 + \ldots + a_n x_n, + \varepsilon Y$$

Now if we have L such observations Y_i, $i = 1,2,\ldots L$, and associated with the *ith* observation is a set of variables x_{ji}, $J = 1,2,\ldots n$

Then

$$Y_i = a_1 x_{1i} + a_2 x_{2i} + \ldots + a_n x_{ni} + \varepsilon Yi, \quad i = 1, \ldots L \qquad (2)$$

Where

εYi is the error associated with the *ith* observation. The sequence of errors εYi, $i = 1,2,\ldots L$ are assumed to have the following statistical properties,

1) The mean of the errors is zero.

2) The errors are sequentially uncorrelated and have a constant variance σ^2

3) The errors are independent of the variables $x_{j,i}$.

The problem then is one of estimating the values of the unknown parameters a_j, $j = 1,2,\ldots n$ given the information $x_{1,i}$, $x_{2,i}$ $\ldots x_{ni}$ and the observation Yi where $i = 1,2,\ldots L$. A simple example to illustrate this problem is that of estimating the position and velocity of an object moving in a straight line with a constant velocity V. From basic physics, we know that the distance d_i at any time t_i is given by:

$$d_i = d_o + vt_i$$

Where d_o is the initial position at t_o and all subsequent distance are measured from a known point.

The linear model of this position consists of L observations of Y_i represent'g

. . ./3

d_i, and error εYi with the statistical properties above, is:

$$Y_i = d_o x_{1i} + v x_{2i} + \varepsilon Yi; \quad i = 1,2,\ldots L$$

Which is in the form of equation (2) where $a_1 = d_o$, $a_2 = V$, x_{1i} 1.0, and $x_{2i} = t_i$. Least Squares analysis, in which the estimate a_j; $j = 1,2,\ldots n$ of n unknown parameters, are chosen to minimize the Least Squares Cost function J where:

$$J = \sum_{i=1}^{L} \left(\sum_{J=1}^{n} x_{ji} \; a_j - Y_1 \right)^2 \tag{3}$$

The minimization is accomplished by setting all the n partial derivates of J with respect to a_j; $j = 1,2,\ldots n$ simultaneously to zero yielding a set of simultaneous equation that are subsequently solved for a_j.

The classical approach to solving for a_j in the multi-parameter case is accomplished by matrix algegra. Rewriting equation (1) in vector-matrix notation we have:

$$Y = X^T a \tag{4}$$

Where

$$X^T = \left(x_1 \; x_2 \; \ldots \; x_n \right) \quad \text{and} \quad a = \left(a_1 \; a_2 \; \ldots \; a_n \right)^T$$

Then

$$D\,\hat{a}\,(J) \simeq \left(\sum_{i=1}^{L} x_i \; x^T \right) \hat{a} - \sum_{i=1}^{L} x_i \; Y_i = 0 \tag{5}$$

and

$$\underline{\hat{a}} = \left[XX^T \right]^{-1} \quad XY \tag{6}$$

Where

$\underline{\hat{a}}$ is the vector of parameter estimates,

and the variance of the parameter estimates is:

$$\text{VAR} \left(\underline{\hat{a}} \right) = \left[XX^T \right]^{-1} \sigma^2$$

Solving the Least Squares problem on micro computers using this classical method or a Singular Value decomposition approach, can be very inefficient, slow, or impractical if the data set is large and multi-parameters need to be estimated. This is because the XX^T matrix (nxn) needs to be inverted (a feat in itself).

The Recursive Least Squares Algorithm is beautifully suited for micro-computers as there is no inversion necessary; and consequently is much faster than the Classical Approaches.

The Recursive Least Squares Algorithm

From equation (6), if the XX^T matrix is non-singular, the solution for $\hat{a}$ can be written.

$$\hat{a}_L = P_L \; b_L \tag{7}$$

Where

$$P_L = \left(\sum_{i=1}^{L} x_i \; x_i^T \right)^{-1} \qquad \text{and } b_L \quad \sum_{i=1}^{L} x_i \; Y_i \tag{8}$$

The recursive form of equation (7), where we wish to estimate $\hat{a}$ after L sampling intervals can be obtained as a linear sum of the estimates obtained after L - 1 samples $\left(\hat{a}_{L-1}\right)$, plus a correction term based on the information Y_L and X_L received during the L th sampling term. From equation (8), P_L and b_L can be related to their prior values by:

$$P_L^{-1} = P_{L-1}^{-1} + X_L \; X_L^T \tag{9}$$

$$b_L = b_{L-1} + X_L \; Y_L \tag{10}$$

The recursive form for P_L in term of P_{L-1} then is:

$$P_L = P_{L-1} - P_{L-1} \; X_L \left(1 + X_L^T \; P_{L-1} \; X_L \right)^{-1} X_L^T \; P_{L-1} \tag{11}$$

. . . /5

and the recursive form for the parameter estimates $\hat{a}$ in terms of $\hat{a}_{L-1}$ is found by substituting equation (11) and (10) into 7 to yield.

$$\hat{a}_L = \hat{a}_{L-1} - P_L \left(X_L \; X_L^T \; \hat{a}_{L-1} - X_L \; Y_L \right) \quad . \tag{12}$$

Equations (11) and (12) form the Recursive Least Squares Algorithm. However, in this form, the parameter estimates are completely deterministic. To incorporate additional information regarding the nature of the observation errors, equation (11) and (12) can be rewritten as:

$$P_L = P_{L-1} - P_{L-1} \; X_L \left(\sigma^2 + X_L^T \; P_{L-1} \; X_L \right)^{-1} X_L^T \; P_{L-1} \tag{13}$$

and

$$\hat{a}_L = \hat{a}_{L-1} - \frac{P_L}{\sigma^2} \left(X_L \; X_L^T \; \hat{a}_{L-1} - X_L \; Y_L \right) \tag{14}$$

When the data variance σ^2 is equal to 1, equations (13) and (14) are equivalent to equations (11) and (12).

The program RLS - II implements the algorithm in equations (13) and (14).

This is an extremely elegant and powerful algorithm as it not only supplies the parameter estimates at each sampling interval, but also presents the accuracy of the estimates in the variance-co-variance matrix P_L. This algorithm is particularly useful for the processing of on-line data where additional information is continually accumulated. Further, this algorithm has several advantages over the classical Least Squares method. Not only does the algorithm provide the parameter estimates at each interval with the minimum of computer storage, the term $\left(\sigma^2 + X_L^T \; P_{L-1} \; X_L \right)^{-1}$ is a scaler quantity requiring no matrix inversion, whereas the classical solution would require an inversion of an L X L matrix at each step.

The Program

Listing A presents the program RLS - II implementing the Recursive Least Squares Algorithm from equations (13) and (14). It is written in Applesoft Basic and will run on a 48K machine. The program is straight forward and easily modified for greater efficiency; I'll suggest a few modifications later on in the article.

Lines 10 to 150 print out the introduction to the program, Line 160 causes a jump to sub-routine 2230 (line 2230 to 2300). This sub-routine allows the user to change the decimal place of the program output. (This rounding only works for numbers > = 1).

Line 170 begins the data entry routines. The user may input data from the keyboard or from a disk file. Lines 1880 to 2020 comprise the disk reading portion of the program. The program allows the user to select disk drive, and to view what files are currently stored on that disk. The data file structure should be obvious from lines 1950 to 1980. Lines 170 to 350 comprise the data entry from keyboard section. In both data entry modes, M is the number of parameters to be estimated, and N is the number of observations. In the keyboard entry modes, these values are entered first. The independent variable(s) X, are then entered (lines 280-300), on a row by column basis. The dependent varibale Y, is entered next (line 350). Lines 390 to 410 find the transpose of the matrix X. At this point, if the entry was from keyboard, the program jumps to line 2060 (at line 420). Lines 2060 to 2190 comprise the data saving routine. This allows the data input from keyboard to be saved to disk. The user can select drive and file name for this data. (The user may also opt to not save data). The program then returns to line 430.

Lines 430 to 600 comprise the algorithm initialization routine. In line 470, the user is asked for an initial value for the P-matrix. (P_{L-1} of equation 13 where L = 1)

The P matrix is the estimation error co-variance matrix and give an indication of the accuracy of the parameter estimates. In practice, usually very little is known about the statistical nature of the parameter. Consequently, the P matrix

. . ./7

is set to large numbers on the diagonal indicating little confidence in the initial parameter estimates, and zero in the other elements indicating no knowledge of the cross variance components of the estimates. The input for the initial P should be in the order of 10^3 to 10^6.

The program then asks for the variance of the observations (σ^2) (line 540). If no knowledge of this variance is known, a 1 is input, changing the algorithm to a deterministic mode (equations 11 and 12). As no a priori knowledge of the parameters exists, the initial parameter vector $\underline{a}$ is set to zero. This is done in lines 590 to 600. Lines 610 to 620 allows the user to pass the output to a printer.

The algorithm resides from lines 730 to 1510, and the new values for P and $\underline{a}$ are assigned and printed in lines 1550 to 1770. The iteration loops at line 1810. In this way, as each piece of information is used, the user can observe the parameter estimates (vector $\underline{a}$), and the measure of their accuracy (matrix P). The computational details of the RLS algorithm are presented in REM statements between lines 630 and 1810.

Two Examples

To demonstrate the use of the RLS program two examples are presented. The first application is the simple two parameter estimation of the moving body problem presented earlier. Table 1 presents some hypothetical data for this problem, Listing B presents the data entry, and the first and last iterations of the output. Table 2 presents the parameter estimate from the RLS- II program, and a program using the classical approach.

The second example is a real life 3 parameter estimation problem from Fisheries Management Biology.

In Fisheries Management, it is often necessary to estimate the optimum stock size N*, an optional sustainable catch C*, an optional catch per effort U*, and an optional total effort E*, for a fish stock of interest. Compounding the problem, the manager often only has two pieces of information, the actual landed catch C_t, and a measure of effort employed to achieve the catch E_t. A simple solution to this problem is to solve the deterministic Schaefer Model:

. . ./8

$$\frac{U_{t+1}}{U_t} - 1 = r - \frac{r}{kq}\left(U_t\right) - qE_t \qquad (15)$$

Where U_t is the Catch per Unit Effort at time t (i.e. C_t/E_t), r is the parameter of intrinsic population growth, K is the carrying capacity (or the population size where net growth is zero), and q is a scaling factor. Equation (15) is a 3 parameter estimation in the form:

$$Y = a_1x_1 + a_2x_2 + a_3x_3 \qquad (16)$$

Where

$$Y = \frac{U_{t+1}}{U_t} - 1$$

$X_1 = 1$

$X_2 = U_t$

$X_3 = E_t$

$a_1 = r$

$a_2 = \frac{-r}{kq}$

$a_3 = q$

K is easily solved once r,q, and a_2 are known.

Then the optimal sustainable catch is:

$$C^* = rk/4$$

the optimum Catch per effort is:

$$U^* = kq/2$$

the optimum effort is

$$E^* = r/2q$$

and the optimum stock size is:

$$N^* = K/2$$

. . ./9

This Schaefer model, although Biologically simple in its assumption, is still used in certain fishery applications.

Table 3 presents the Catch and Effort Data for the Hecate Strait Rock Sole. It's analysis, using the RLS II program and the classical approach, are presented in Listing C.

Program Improvement

Program RLS II is a reasonable program for Recursive Least Squares Analysis, but is not memory efficient. First of all, nearly 40% of the program is REM statements; these can be easily removed without affecting program operation. Secondly, as the algorithm is sequential, data can be analyzed on line, (i.e.) there is no need to read the entire data set at once. The program can easily be modified to accept input at each observation step and then present the up-dated parameter estimates($\hat{a}$) and the error-covariance matrix P. This modification would also improve memory requirements. The computational portion of the program was written to explicitly show how the calculations are done for reasons of clarity. Some of these steps can be combined to speed up the algorithm. For example, the calculation in line 960 could be incorporated in the loop between lines 890 and 900 with line 870 changed to FOR I = 1 TO M: W5(I,1) = 0: W6(I,1)=0 and thus eliminating lines 940 to 970.

Conclusion

The RLS algorithm is an extremely important and elegant tool for the data analyst. In more complex situations, the RLS algorithm has been developed to track parameters that vary through time, and the filtering of noisy input data for optimal estimation and control. Such algorithms were used to determine in-flight trajectory calculations for the Apollo missions, and are used to provide parameter up-dates and control output for cruise missile guidance. Their minimal memory requirement and efficiency make these algorithm suitable for small machine use.

References

BOX, E.P., and G.M. Jenkins (1976) Time Series Analysis: Forcasting and Control, Holden-Day.

Gelb, A., et at (1974) Applied Optimal Estimation, MIT Press.

Young, P. (1974) Recursive Approaches to Time Series Analysis. Bull. of Inst. Math. Applications 10 (5/6) 204-224.

ACKNOWLEDGEMENTS

I wish to express thanks to Dr. Ray Hilborn, Institute of Animal Resource Ecology, U.B.C. for introducing the Recursive Least Square Algorithm to me.

Listing A

```
LIST

10  REM  ***** PROGRAM RLSII *****
20 D$ =  CHR$ (4):D = 5
30  HOME : FOR I = 0 TO 39: PRINT "0";: NEXT : PRINT "0";: HTAB 39: PRINT " 0";: PRINT "0
    RLS-II";: HTAB 39: PRINT " 0";
40  PRINT "0";: HTAB 39: PRINT " 0";: PRINT "0              A PROGRAM FOR";: HTAB 39: PRINT "
    0";: PRINT "0   RECURSIVE LEAST SQUARES ANALYSIS ";: HTAB 39: PRINT " 0";
50  PRINT "0              OF REAL DATA.  ";: HTAB 39: PRINT " 0";: PRINT "0";: HTAB 39: PRINT
    " 0";: PRINT "0     ";: HTAB 39: PRINT " 0";
60  PRINT "0           COPYRIGHT (C) 1983";: HTAB 39: PRINT " 0";: PRINT "0";: HTAB 39: PRINT
    " 0";: PRINT "0                    BY";: HTAB 39: PRINT " 0";
70  PRINT "0";: HTAB 39: PRINT " 0";: PRINT "0              R. V. PALERMO ";: HTAB 39: PRINT "
     0";: PRINT "0  ";: HTAB 39: PRINT " 0";: PRINT "0         BIOMETRICIAN, PROGRAMMER ";: HTAB
    39: PRINT " 0";
80  PRINT "0         DEPT. FISHERIES AND OCEANS";: HTAB 39: PRINT " 0";
90  PRINT "0         VANCOUVER, CANADA,  MAY/83";: HTAB 39: PRINT " 0";
100  PRINT "0";: HTAB 39: PRINT " 0";: FOR I = 0 TO 39: PRINT "0";: NEXT
110  VTAB 22: PRINT "PLEASE PRESS RETURN": INPUT " ";Q$
120  REM  ----------------------------------------
130  HOME : REM   INTRODUCTION
140  REM    RECURSIVE LEAST SQUARES PARAMETER ESTIMATES FOR TIME SERIES DATA
150  REM  ----------------------------------------
160  GOSUB 2230
170  INVERSE : HOME : PRINT "DATA ENTRY": NORMAL : VTAB (10): INPUT "IS DATA ON FILE (Y/N) ?";
    RS$: IF RS$ = "Y" THEN 1800
180  HOME : INVERSE : PRINT "INPUT DATA FROM KEYBOARD": NORMAL : PRINT
190  INPUT " ROWS (0 PARMS ) :";M
200  INPUT " COLUMNS (0 DATA POINTS ) :";N
210  REM  DIMENSION WORK SPACE BUFFERS
220  DIM X(N,N),Y(N)
230  DIM XT(N,N),P(N,N),A(N,N),W1(N,1),W2(1,N),W3(N,1),W4(N,N)
240  DIM W5(N,1),W6(1,N),W8(N,1),W9(N,N),PN(N,N)
250  DIM T1(N,N),T2(N,1),T3(N,1),T4(N,1),T5(N,1),T6(N,1)
260  IF F2 = 1 THEN 370
270  PRINT : INVERSE : PRINT "INPUT DATA ITEMS ": NORMAL : PRINT
280  FOR I = 1 TO N: FOR J = 1 TO N
290  PRINT "X(";I;",";J;") = ";: INPUT X(I,J)
300  NEXT : NEXT
310  REM  ----------------------------------------
320  REM  INPUT Y DATA
330  REM  ----------------------------------------
340  PRINT "INPUT Y DATA"
350  FOR I = 1 TO N: PRINT "Y(";I;") = ";: INPUT Y(I): NEXT
360  REM  ----------------------------------------
370  REM   TRANSPOSE X
380  REM  ----------------------------------------
390  FOR I = 1 TO N: FOR J = 1 TO N
400 XT(J,I) = X(I,J)
410  NEXT : NEXT
420  IF F2 = 0 THEN  GOSUB 2060
430  REM  ----------------------------------------
440  REM  SET P0 MATRIX INITIALIZATION
450  REM  ----------------------------------------
460  HOME : VTAB 3: INVERSE : PRINT " ALGORITHM INITIALIZATION ": NORMAL : VTAB 5
470  INPUT " INPUT P-MATRIX INITIAL VALUE : ";P9
```

```
480  FOR I = 1 TO N: FOR J = 1 TO N
490  IF J = I THEN P(I,J) = P9: GOTO 510
500 P(I,J) = 0
510  NEXT
520  NEXT
530  PRINT : PRINT
540  INPUT "INPUT VARIANCE = ";V2
550  PRINT : PRINT
560  REM  ---------------------------------------
570  REM  SET PARM MATRIX A TO 0
580  REM  ---------------------------------------
590  FOR I = 1 TO N: FOR J = 1 TO N
600 A(I,J) = 0: NEXT : NEXT
610  PRINT : PRINT : INPUT "HARD OUTPUT TO PRINTER (Y/N)?";RS$
620  IF RS$ = "Y" THEN  PRINT : INPUT "PRINTER SLOT= ";DU: PRINT D$;"PR#";DU
630  REM  ---------------------------------------
640  REM  LOOP OVER ALL DATA TO N AND BEGIN RECURSIVE ESTIMATION
650  REM  ---------------------------------------
660  REM    M1=X,M2=XT, M3= A(N,L),M4=P
670  REM   M5=PX, M6= XT*P, M7=(V2+ XT*P*X)^-1
680  REM  M8= M5 * M7, M9= M9* M6 , PN = NEW P MATRIX AT K
690  REM  ---------------------------------------
700  REM
710  REM   ****************************************
720  HOME : INVERSE : FLASH : PRINT "WORKING": NORMAL
730  FOR L = 1 TO N
740  REM
750  REM  ---------------------------------------
760  REM  ASSIGN DATA TO WORK SPACE BUFFER FOR THIS ITERATION
770  REM  ---------------------------------------
780  FOR I = 1 TO N
790 M1(I,1) = X(I,L):M2(1,I) = XT(L,I)
800  NEXT
810  FOR I = 1 TO N: FOR J = 1 TO N
820 M4(I,J) = P(I,J): NEXT : NEXT
830  REM  ---------------------------------------
840  REM  PX=M4-M5*(M7)*M6
850  REM  CALCULATE M5= M4*M1
860  REM  ---------------------------------------
870  FOR I = 1 TO N:M5(I,1) = 0
880  FOR J = 1 TO N
890 M5(I,1) = M4(I,J) * M1(J,1) + M5(I,1)
900  NEXT : NEXT
910  REM ---------------------------------------
920  REM  CALCULATE M6=M2*M4
930  REM ---------------------------------------
940  FOR I = 1 TO N:M6(1,I) = 0
950  FOR J = 1 TO N
960 M6(1,I) = M2(1,J) * M4(J,I) + M6(1,I)
970  NEXT : NEXT
980  REM ---------------------------------------
990  REM   CALCULATE M7=(V2+M6*M1)^-1
1000  REM  ---------------------------------------
1010 M7 = 0: FOR I = 1 TO N
1020 M7 = M7 + M6(1,I) * M1(I,1)
1030  NEXT :M7 = M7 + V2:M7 = 1 / M7
1040  REM ---------------------------------------
1050  REM  CALCULATE M8=M5*M7
```

```
1950  REM ----------------------------------------
1970  FOR I = 1 TO N:M8(I,1) = M5(I,1) * M7: NEXT
1980  REM ----------------------------------------
1990  REM  CALCULATE M9=M8*M6
1100  REM ----------------------------------------
1110  FOR I = 1 TO N: FOR J = 1 TO N
1120 M9(I,J) = M8(I,1) * M6(1,J)
1130  NEXT : NEXT
1140  REM  ----------------------------------------
1150  REM  CALCULATE NEW P MATRIX IN PN=M4-M9
1160  REM  ----------------------------------------
1170 K9 =  - 1
1180  FOR I = 1 TO N: FOR J = 1 TO N
1190 PN(I,J) = M4(I,J) + (K9 * M9(I,J))
1200  NEXT : NEXT
1210  REM  ----------------------------------------
1220  REM  CALCULATE T1=M1*M2
1230  REM  ----------------------------------------
1240  FOR I = 1 TO N: FOR J = 1 TO N
1250 T1(I,J) = M1(I,1) * M2(1,J)
1260  NEXT : NEXT
1270  REM   ----------------------------------------
1280  REM  CALCULATE T2=T1*M3
1290  REM ----------------------------------------
1300  FOR I = 1 TO N:T2(I,1) = 0
1310  FOR J = 1 TO N
1320 T2(I,1) = T1(I,J) * M3(J,1) + T2(I,1)
1330  NEXT : NEXT
1340  REM  ----------------------------------------
1350  REM  CALCULATE T3=M1*Y(L)
1360  REM  ----------------------------------------
1370  FOR I = 1 TO N:T3(I,1) = M1(I,1) * Y(L): NEXT
1380  REM   ----------------------------------------
1390  REM  CALCULATE T4=T2-T3
1400  REM  ----------------------------------------
1410  FOR I = 1 TO N:T4(I,1) = T2(I,1) + (K9 * T3(I,1)): NEXT
1420  REM  ----------------------------------------
1430  REM  CALCULATE T5=PN*T4
1440  REM  ----------------------------------------
1450  FOR I = 1 TO N:T5(I,1) = 0
1460  FOR J = 1 TO N:T5(I,1) = (PN(I,J) / V2) * T4(J,1) + T5(I,1): NEXT : NEXT
1470  REM   ----------------------------------------
1480  REM  CALCULATE NEW PARMS IN T6=M3-T5
1490  REM ----------------------------------------
1500  FOR I = 1 TO N:T6(I,1) = M3(I,1) + (K9 * T5(I,1)): NEXT
1510  PRINT : PRINT
1520  REM  ----------------------------------------
1530  REM   ASSIGN NEW PARMS TO A
1540  REM  ----------------------------------------
1550  FOR I = 1 TO N:A(I,L) = T6(I,1):M3(I,1) = T6(I,1): NEXT
1560  REM  ----------------------------------------
1570  REM  ASSIGN NEW P MATRIX
1580  REM  ----------------------------------------
1590  FOR I = 1 TO N: FOR J = 1 TO N
1600 P(I,J) = PN(I,J): NEXT : NEXT
1610  REM  ----------------------------------------
1620  REM  PRINT OUT THESE NEW VALUES FOR NEXT ITERATIONS
1630  REM ----------------------------------------
```

```
1640  HOME : PRINT : PRINT
1650  PRINT " PARMS ESTIMATES AT ITERATION ";L
1660  PRINT
1670  FOR I = 1 TO M
1680  IF A(I,L) < 1 THEN  PRINT A(I,L); SPC( 3): GOTO 1700
1690  PRINT  INT (A(I,L) * P1 + .5) / P1; SPC( 3)
1700  NEXT : PRINT
1710  PRINT : PRINT
1720  PRINT "NEW P MATRIX "
1730  FOR I = 1 TO M: FOR J = 1 TO M
1740  IF P(I,J) < 1 THEN  PRINT P(I,J); SPC( 3): GOTO 1760
1750  PRINT  INT (P(I,J) * P1 + .5) / P1; SPC( 3);
1760  NEXT : PRINT
1770  NEXT
1780  REM  ------------------------------------ ---
1790  REM  END ITERATION LOOP
1800  REM  ----------------------------------------
1810  NEXT L
1820  PRINT D$;"PR#0"
1830  VTAB 21: PRINT  CHR$ (7): INVERSE : FLASH : PRINT "FINISHED": NORMAL : END
1840  REM **********************************
1850  REM  ----------------------------------------
1860  REM  DATA INPUT FROM DISK ROUTINE
1870  REM  ----------------------------------------
1880  DIM A1(50,5),Y(50)
1890 F2 = 1
1900  HOME : INVERSE : PRINT "READ DATA FROM DISK ROUTINE": NORMAL
1910  VTAB 5: PRINT : INPUT "INPUT DRIVE NUMBER TO READ DATA : ";DR: PRINT : INVERSE : PRINT "
     CURRENT FILES ON DISK": NORMAL
1920  PRINT D$;"CATALOG,D";DR: PRINT
1930  INPUT " INPUT FILE NAME ";FS$
1940  PRINT : PRINT "READING DATA FILE : ";FS$
1950  PRINT D$;"OPEN";FS$: PRINT D$;"READ";FS$: INPUT M: INPUT N
1960  FOR I = 1 TO M: FOR J = 1 TO N
1970  INPUT A1(I,J): NEXT : NEXT
1980  FOR I = 1 TO M: INPUT Y(I): NEXT
1990  PRINT D$;"CLOSE";FS$
2000  DIM X(N,M): FOR I = 1 TO M: FOR J = 1 TO N:X(J,I) = A1(I,J): NEXT : NEXT
2010 A5 = M:M = N:N = A5
2020  GOTO 230
2030  REM  ----------------------------------------
2040  REM  SAVE DATA TO DISK ROUTINE
2050  REM  ----------------------------------------
2060  PRINT : PRINT : INPUT "SAVE DATA TO DISK (Y/N)? ";RS$
2070  IF RS$ = "N" THEN 2190
2080  HOME : INVERSE : PRINT "SAVE DATA TO DISK ROUTINE": NORMAL
2090  VTAB 5: PRINT : INPUT "INPUT DRIVE NUMBER TO SAVE DATA : ";DR: PRINT : INVERSE : PRINT "
     CURRENT FILES ON DISK": NORMAL
2100  PRINT D$;"CATALOG,D";DR: PRINT
2110  INPUT " INPUT FILE NAME TO SAVE: ";FS$
2120  PRINT : PRINT "SAVEING DATA FILE : ";FS$
2130  PRINT D$;"OPEN";FS$: PRINT D$;"WRITE";FS$: PRINT N: PRINT M
2140  FOR I = 1 TO N: FOR J = 1 TO M
2150  PRINT XT(I,J): NEXT : NEXT
2160  FOR I = 1 TO N: PRINT Y(I): NEXT
2170  PRINT D$;"CLOSE";FS$
2180  HOME
2190  RETURN
```

```
2200  REM  ------------------------------------
2210  REM  CHANGE DECIMAL ROUTINE
2220  REM  ------------------------------------
2230  HOME : INVERSE : PRINT "CURRENT SYSTEM DECIMAL PLACES  ": PRINT " FOR OUTPUT IS ":: NORMAL
    : PRINT "  ";D
2240 P1 = 10 ^ D
2250  VTAB 4: INPUT "DO YOU WISH TO CHANGE THIS VALUE (Y/N)?";RS$
2260  IF RS$ = "N" THEN 2300
2270  VTAB 4: CALL  - 958
2280  VTAB 8: INPUT "INPUT NEW DECIMAL PLACES FOR OUTPUT :";D
2290  GOTO 2230
2300  RETURN
```

TABLE 1

SAMPLE DATA FOR THE MOVING BODY PROBLEM

RAW DATA							
	OBSERVED DISTANCE IN FEET (Y)	9	18	23	31	52	87
	TIME T	1	2	3	4	8	12

SAMPLE	X1	X2	Y
1	1	1	9
2	1	2	18
3	1	3	23
4	1	4	31
5	1	8	52
6	1	12	87

TABLE 2

RESULTS FOR THE ABOVE PROBLEM

EQUATION: Y= A1 X1 +A2 X2

PARAMETER ESTIMATES		RLSII	SVD (CLASSICAL SOLUTION)
INITIAL POSITION	A1	2.68939	2.68939
VELOCITY	A2	6.79545	6.79545

Listing B

RLS OUTPUT FOR MOVING BODY PROBLEMS

```
 ]PR#0DATA ENTRY
 IS DATA ON FILE (Y/N) ?N
 INPUT DATA FROM KEYBOARD

 ROWS (# PARMS ) :2
 COLUMNS (# DATA POINTS ) :6

INPUT DATA ITEMS

X(1,1) = ?1
X(1,2) = ?1
X(1,3) = ?1
X(1,4) = ?1
X(1,5) = ?1
X(1,6) = ?1
X(2,1) = ?1
X(2,2) = ?2
X(2,3) = ?3
X(2,4) = ?4
X(2,5) = ?8
X(2,6) = ?12
INPUT Y DATA
Y(1) = ?9
Y(2) = ?18
Y(3) = ?23
Y(4) = ?31
Y(5) = ?52
Y(6) = ?87


SAVE DATA TO DISK (Y/N)? N
 ALGORITHM INITIALIZATION
 INPUT P-MATRIX INITIAL VALUE : 1E6


INPUT VARIANCE = 1



HARD OUTPUT TO PRINTER (Y/N)?Y

PRINTER SLOT= 1
WORKING



 PARMS ESTIMATES AT ITERATION 1

4.5  4.5


NEW P MATRIX
500000.25   -499999.75
-499999.75   500000.25

 PARMS ESTIMATES AT ITERATION 2

0  9


NEW P MATRIX
5  -3
-3  2


 PARMS ESTIMATES AT ITERATION 3

2.66667  7


NEW P MATRIX
2.33333  -1
-1  .5


 PARMS ESTIMATES AT ITERATION 4

2.5  7.1


NEW P MATRIX
1.5  -.5
-.5  .2


 PARMS ESTIMATES AT ITERATION 5

5  6


NEW P MATRIX
.643835615   -.123287671
-.123287671   .0342465754


 PARMS ESTIMATES AT ITERATION 6

2.68939  6.79545


NEW P MATRIX
.450757575   -.0568181817
-.0568181817   .0113636364
```

TABLE 3

EFFORT DATA FOR THE HECATE ROCK SOLE

YEAR	CATCH	X3 EFFORT	X2 UT= CATCH/EFFORT	Y (UT+1/UT)-1
1954	835	827	1.009674	-.267012
1955	1324	1789	.7400783	-.149054
1956	512	813	.6297663	-.174766
1957	211	406	.5197044	1.039314
1958	797	752	1.059840	-.214919
1959	109	131	.8320611	.7786019
1960	626	423	1.479905	.0468884
1961	660	426	1.549296	-.237879
1962	503	426	1.180751	-.135891
1963	754	739	1.020298	-.245833
1964	484	629	.7694754	1.006158
1965	318	206	1.543689	-.462426
1966	1468	1769	.8298474	.0246279
1967	1045	1229	.8502848	-.470766
1968	1386	3000	.45	.7332488
1969	1230	1577	.7799620	-.435820
1970	455	1034	.4400387	-.113764
1971	895	2295	.3899782	-.281636
1972	230	821	.2801462	.6432781
1973	180	391	.4603581	-.305394
1974	220	688	.3197674	1.001739
1975	562	878	.6400911	-.140844
1976	892	1622	.5499383	-.345320
1977	427	1186	.3600337	.2227284
1978	394	895	.4402235	-.045939
1979	420	1000	.42	

X1 = 1 FOR ALL YEARS

Listing C

RLS OUTPUT FOR PROBLEM 2 (Schaefer Model)

```
IFR00 ALGORITHM INITIALIZATION
 INPUT P-MATRIX INITIAL VALUE : 1E6

 INPUT VARIANCE = 1

 WORKING


 PARMS ESTIMATES AT ITERATION 1

 0   -1.2207031E-04   -.0407457352

 NEW P MATRIX
 999998.538   -1.47627976   -1209.18627
 -1.47627976   999998.589   -1220.88336
 -1209.18627   -1220.88336   2.95288

 PARMS ESTIMATES AT ITERATION 2

 -24.59375   -30.5040283   .0263288869

 NEW P MATRIX
 606547.03   -488446.552   -136.980825
 -488446.552   393624.949   110.19165
 -136.980825   110.19165   .0309845172
                 .
                 .
                 .
                 .
                 .
 PARMS ESTIMATES AT ITERATION 25

 .758076288   -.657671854   -1.95391222E-04

 NEW P MATRIX
 .47573414   -.353234888   -1.66020982E-04
 -.353234217   .360335325   0.05372391E-05
 -1.66021379E-04   0.05373922E-05   1.0434239E-07
```

RESULTS FOR THE ABOVE PROBLEM

EQUATION: Y= A1 X1 +A2 X2 +A3 X3

PARAMETER ESTIMATES		RLSII	SVD (CLASSICAL SOLUTIO
INTRINSIC GROWTH RATE	A1	.75807	.76353
-R/KQ	A2	-.65767	-.65722
SCALING FACTOR Q	A3	-1.95E-4	-2.02E-4
VALUES			
CARRYING CAPACITY	K	5899.279	5759.142
OPTIMAL CATCH	C*	1118.017	1099.319
OPTIMAL C/E	U*	.5763301	.5808785
OPTIMAL EFFORT	E*	1939.889	1892.512
OPTIMAL POPULATION	X*	2949.648	2879.571

POOR

QUALITY

ORIGINAL